The hyperwall: A multiple flat-panel display wall for high-dimensional visualization

Chris Henze, Creon Levit, David Ellsworth, and Timothy Sandstrom*
NASA Advanced Supercomputing Division
NASA Ames Research Center
Moffett Field CA 94035 USA

1 Basic Concept

The NASA Ames "hyperwall" consists of a 7 × 7 matrix of flat panel displays driven by 49 rackmounted dual-CPU nodes, each with its own high-end graphics card. It is similar to so-called "powerwalls" in that a cluster of computers drives a cluster of displays. However, the hyperwall's visualization utility, and hence its design philosophy and system architecture, are fundamentally different from those of a powerwall.

In powerwalls, a single rendering task is spread across multiple processors and then the multiple outputs are tiled into a single seamless "superdisplay". In contrast, the hyperwall emphasizes displaying multiple independent - but related - images, and providing useful means for composing and controlling these image sets. In place of elaborate software or hardware crossbar switches, we rely on the human visual system for integration, synthesis, and pattern discrimination in complex and high-dimensional data spaces.

We note that the hyperwall actually subsumes traditional powerwall applications. We could, for example, render a single scene across the hyperwall displays – perhaps using Chromium, or the like. Of course, the composite display of the hyperwall is discontinuous, but we do not consider this to be a serious drawback, and in fact we think the color balance and image quality of the hyperwall's flat panels is in general superior to typical powerwall projected displays. In addition, flat panel displays are considerably cheaper than projectors, the display system requires less volume (it is only 9 × 12 × 3 feet), and the system is far easier to adjust and maintain.

2 Human Factors

We have sized the hyperwall as a 7 × 7 array for several reasons. Some are pedestrian: The total system fit within our budget and the display array fit within our lab. The 7 × 7 18" LCD monitor array is a comfortable size - about the size of an office wall. From the middle of a standard size office the user's gaze can easily fixate on any hyperwall monitor with little or no head movement. When viewing it from this "natural" distance the hyperwall display subtends about 80 degrees, and each pixel subtends about 1/100 degree. This nicely matches both the field of view and resolution of the human eye.

We are additionally motivated by Miller's foundational paper that married experimental psychology with information theory: "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information" [1]. Miller showed that seven is about the number of classes available for absolute univariate discrimination, or the number of objects in the span of attention, or the size of a "chunk" in human short term memory.

On the hyperwall we have two dimensions to work with, simultaneously, so the natural size is about 7 × 7 – the size, according to Miller, of a two dimensional (bivariate) chunk. If the image array is much larger than this we are unable to form a gestalt of the entire information field. If the array is smaller we fall below our maximum perceptual capacity.

3 Hardware

The displays in the hyperwall are 18" Samsung 181T flat panel monitors. Each LCD is framed by a uniform black plastic bezel approximately 3/4" in width. A custom designed mounting rack allows pitch and yaw adjustment of each monitor, as well as translational adjustment between rows and columns. In addition, each monitor can be moved independently up to 14" in "z" - perpendicular to the frame - to allow nonplanar arrangements of the viewing surfaces - for example spherical or paraboloid sections. We designed the rack to provide all these degrees of freedom in order to compensate for the directionality in the monitors, and to accommodate different viewing distances. In practice, the Samsung monitors deliver well on their promised 170 degree omnidirectional field of view, so a wide range of adjustments is effective. Figure 1 shows a mechanical drawing of the mounting system.

Each display is driven directly by an nVidia GeForce 4 Ti4600 graphics card, at 1280 × 1024 resolution. The aggregate pixel count for the entire display matrix is thus 7 × 7 × 1280 × 1024 = 64,225,280 = 64 M pixels.

Each graphics card is housed in a dual-CPU AMD Athlon MP2000+ rackmounted slave node. The slave nodes each have a 100GB IDE disk, thus providing aggregate storage of approximately 5TB. The slaves are driven by a similarly configured master node, and all communication is via Fast (100 BaseT) Ethernet, coordinated by a pair of Cisco Catalyst 2950 G-48-EI switches.

4 Software Infrastructure

We are running a standard Red Hat Linux 7.3 installation on the master and on each of the slave nodes. We use SystemImager (sourceforge.net/projects/systemimager) to maintain consistent slave node configurations. Slave nodes communicate with the master on an internal Ethernet interface, with names and IPs being assigned by the master via DHCP at slave boot time. In addition, the master exports several of its filesystems, which are mounted by the slaves using NFS.

We can run many standalone X Window System event-driven applications virtually unchanged as hyperwall SPMD applications using the simple ploy of replicating mouse and keyboard events on the master and broadcasting them to the slaves. We specialize the input data for each slave node using script-generated symbolic links to data in the globally accessible NFS. The X event-cloning mechanism is remarkably effective at exposing data-dependent behavior in client programs - for example, incremental rotation schemes that

*{chenze,creon,ellswort,sandstro}@nas.nasa.gov

are affected by rendering frame rates cause rapid desynchronization of scenes with differing numbers of polygons.

Applications that are designed to support collaboration are promising hyperwall candidates. We have one such application that exchanges script commands to coordinate multiple interacting views. In the hyperwall instantiation, the operator drives one copy of the analysis tool on the master node, which broadcasts script commands to copies of the program running on each of the slave nodes.

For controlling true MPMD applications, we use a distributed object framework with a robust signal/slot event service, the details of which are outside the scope of this report.

For any of the schemes discussed above, there is the general issue of mapping and distributing datasets onto the slave nodes, either on-demand via NFS or some other global file system scheme, or as a prestaging phase. Determining the layouts and mechanisms of data distribution that support desired access patterns is a subject we are presently investigating.

5 Applications

The tabular layout of the hyperwall supports many existing "multiview" visualization paradigms, including spreadsheet-style approaches [2, 3], multidimensional/multivariate techniques [4, 5], and brushing/linking [6, 7]. Hyperwall versions of these techniques benefit from the dual-CPU and graphics subsystem devoted to each view, and this same feature allows us to extend these techniques from a postprocessing scenario to "concurrent analysis" - i.e., each view is connected directly to an ongoing simulation or other compute-intensive application.

The hyperwall is obviously well-suited for parameter studies. For instance, one can visualize a 2D parameter study of airfoils with 7 different cambers and 7 different thicknesses, all 49 combinations running the same 2D compressible flow code. As the Mach number is increased across all simulations, one can see where in the design space effects such as separation or vortex shedding first occur. One can zoom and pan in such a 2D parameter space by simple reassignment of the slave nodes. Higher-dimensional parameter spaces can be explored using 2D cuts.

Or one can investigate a single airfoil showing combinations of various dynamical variables with various visualization techniques and parameters. For instance, one can plot energy, density, momentum, pressure, temperature, and so forth across the rows of the hyperwall, while distributing streamlines, contours, volume rendering, colormaps and viewing angles across the columns. The resulting combinatoric array of visualizations permit side-by-side comparison and allow the user to choose the most effective means of interrogating and presenting a feature of interest. Figure 2 shows a related example using the Space Shuttle main engine liquid hydrogen turbopump.

We have used the hyperwall to visualize the three dimensional electrostatic potential for a "complete" array of small 2nd row hydride molecules at equilibrium geometry, and the quantum potential for a one-parameter sequence of ethylene molecules undergoing parameterized out-of-plane bond deformation. We have visualized the results of a meso-scale atmospheric dynamics simulation, displaying related 3D scalar fields simultaneously, each on its own display. In all of these cases, changes due to, e.g. viewing angle, transfer function (for volume visualization) or contour level (for isosurfaces) specified interactively by the master are simultaneously and independently computed, rendered, and displayed by the slaves.

The hyperwall shows particular promise for exploring high-dimensional data. A 2D array of "3D" graphics scenes portrays 5 dimensions rather explicitly. We are currently attempting to provide a tool for interactively exploring the 6-dimensional electronic pair density function calculated for small molecules. If we sample this space at 128 points along each side of a 6D hypercube, we will generate 4.4 trillion datapoints - which can just fit on the aggregate disk of the hyperwall, at 8-bit resolution. If we constrain things so that each display reads only from its set of 42,799 2MB datasets, then successive sets of 49 128^3 "slices" can be displayed at about 10 Hz, permitting "structured browsing" of a hitherto intractable dataset.

References

[1] G. A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *The Psychological Review*, 63:81–97, 1956.

[2] M. Levoy. Spreadsheets for images. In Andrew Glassner, editor, *SIGGRAPH 94 Conference Proceedings*, Annual Conference Series, pages 139–146. ACM SIGGRAPH, Addison Wesley, August 1994.

[3] E. H. Chi, J. Riedl, P. Barry, and J. Konstan. Principles for information visualization spreadsheets. *IEEE Computer Graphics and Applications*, 18(4):30–38, July/August 1998.

[4] W. S. Cleveland. *Visualizing Data*. Hobart Press, New Jersey, 1993.

[5] J. J. van Wijk and R. van Liere. Hyperslice – visualization of scalar functions of many variable. In *IEEE Visualization '93*, pages 119–125. IEEE, 1993.

[6] M. O. Ward. Xmdvtool: Integrating multiple methods for visualizing multivariate data. In *IEEE Visualization '94*, pages 326–336. IEEE, October 1994.

[7] C. Henze. Feature detection in linked derived spaces. *IEEE Visualization '98*, pages 87–94, October 1998. ISBN 0-8186-9176-X.

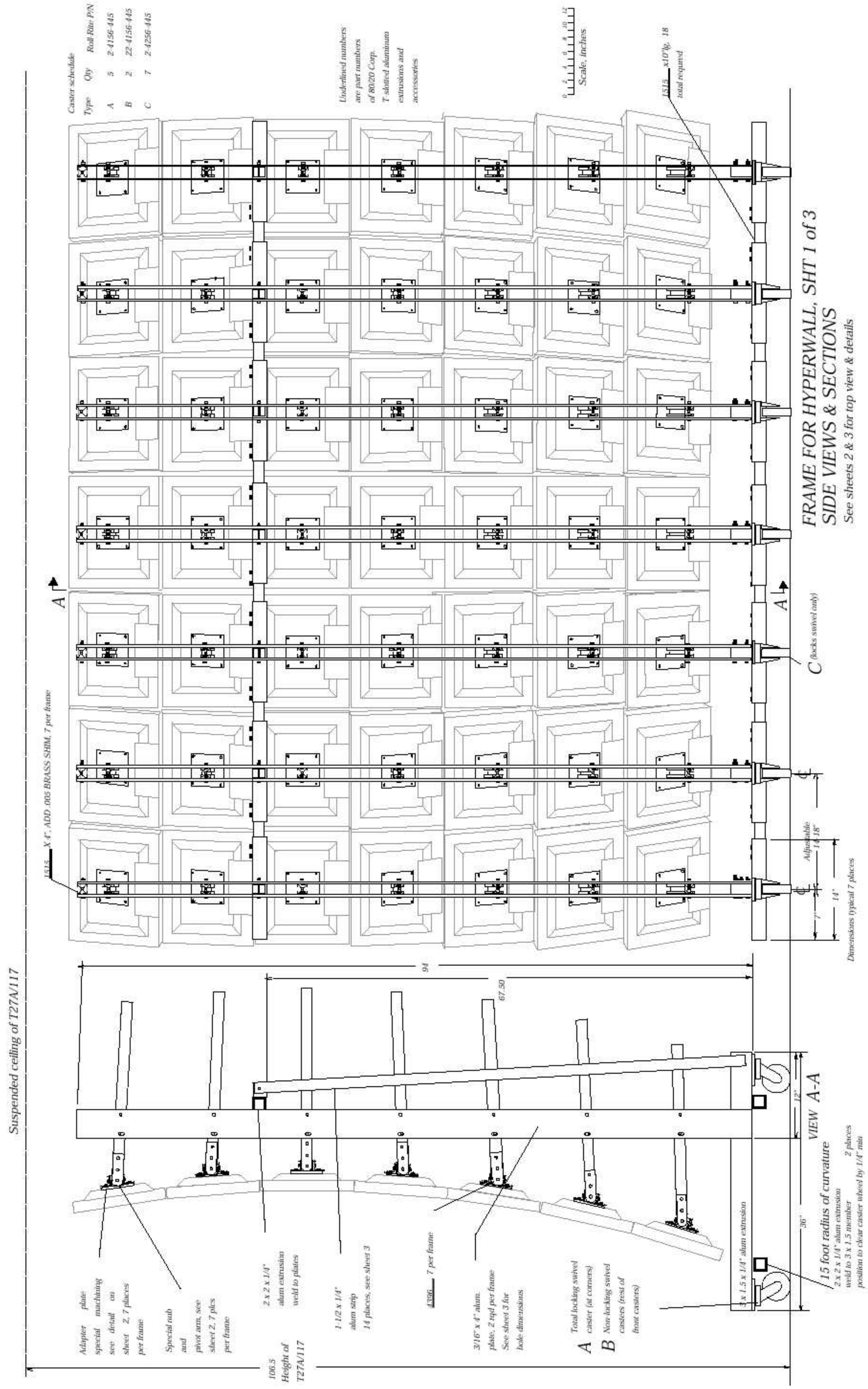

Figure 1: Mechanical drawing of the display mounting system.

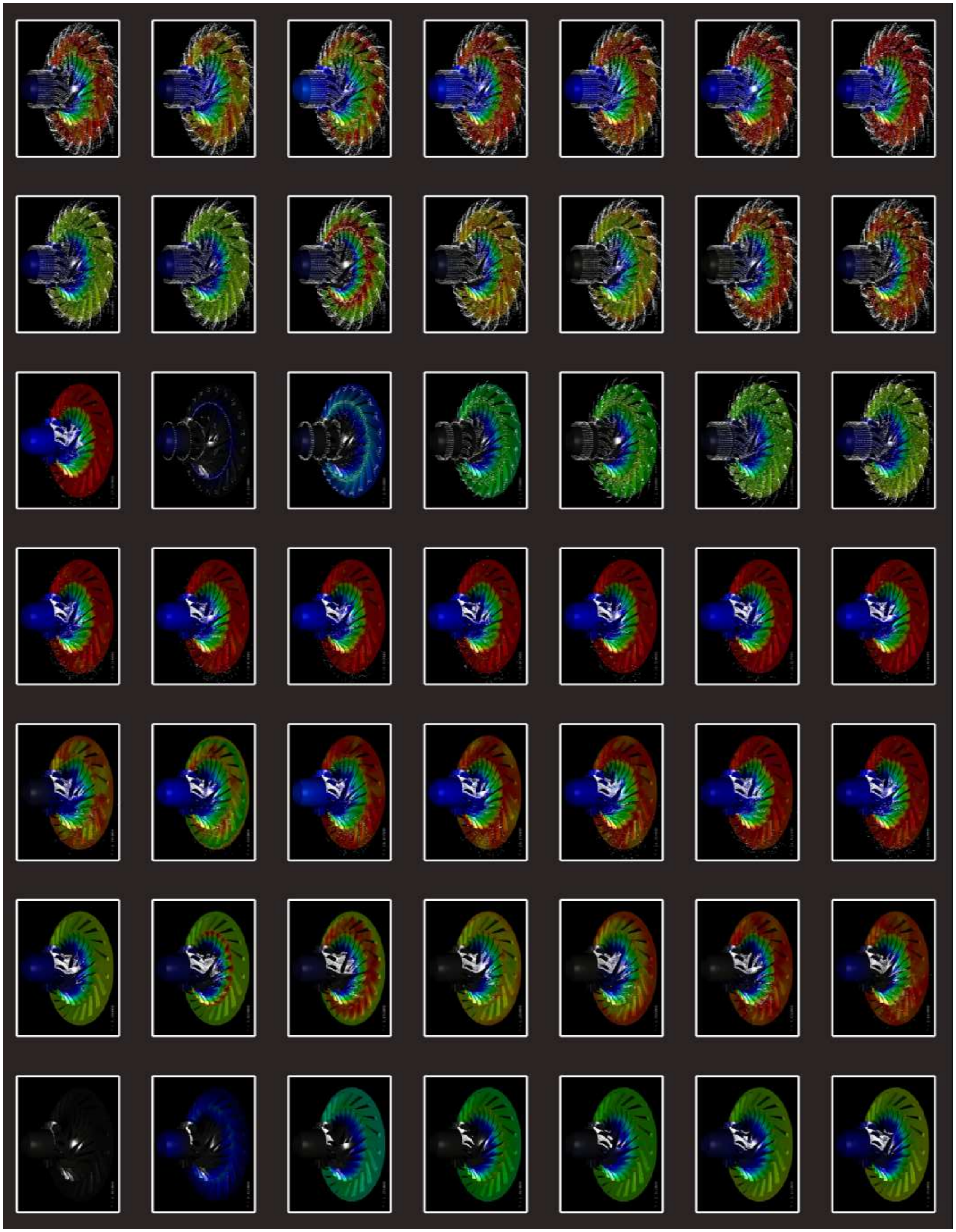

Figure 2: Example of a 7 $\times$ 7 hyperwall display showing different visualizations and timesteps of a Space Shuttle main engine liquid hydrogen turbopump CFD simulation.